CERTIFICATION OF TRANSLATION

I, Elise Duvekot, a citizen of the United States of America, hereby certify that I am fully familiar with the German and English languages and that I am capable of translating from German into English. To the best of my knowledge and ability, the foregoing pages constitute an accurate and complete translation of the text before me in the German language of the following:
German Priority Application No. 101 49 637 titled "Elektronische Paketfachanlage und Logistiksystem".

I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true.

In witness whereof I sign,

August 3, 2007
Date

Elise Duvekot
Signature of translator

Öffentlich bestellte und beeidigte Urkundenübersetzerin und Dolmetscherin
Elise Duvekot
der englischen Sprache
für Baden-Württemberg

Translation by: **Duvekot Translators & Interpreters**
131 Bloor Street West, Suite 803
Toronto, ON M5S 1S3, Canada
Phone: (+1) 647-435-1060
Fax: (+1) 647-438-2978
e-mail: LEDTRANS@CS.COM

PACKET BOX ELECTRONIC DEVICE AND LOGISTIC SYSTEM

Description:

The invention relates to an electronic parcel box system.

The invention is based on the objective of creating an electronic parcel box system that allows flexible delivery and pick-up of parcels from parcel box systems.

According to the invention, this objective is achieved in that the electronic parcel box system is configured in such a way that it comprises a plurality of electronic parcel boxes, whereby in each case, several electronic parcel boxes are associated with an operating unit, and whereby a central control unit is provided for controlling the operating units.

The invention provides that the electronic parcel box system consists of several individual delivery devices, whereby the delivery devices are equipped with one or more operating units. Several or all of the operating units are controlled by the central control unit.

It is advantageous for the operating units to comprise a control means for controlling the opening and/or the closing of the parcel boxes associated with the operating units.

The individual electronic parcel box systems are joined together to form clusters.

The provision of a central control unit considerably increases the functionality and the capabilities of the electronic parcel box system.

In particular, it is advantageous for the central control unit to have a means for associating the parcel box systems with the operating units.

The embodiment in which the parcel boxes and the operating units are associated with each other via the central control unit has the advantage that, in this manner, the electronic parcel box system can easily be adapted to changed operating parameters.

In particular, it is advantageous for the association of the parcel boxes with the operating units to be variable.

In this manner, it is possible, for example, that, if an operating unit malfunctions, parcel boxes that had been associated with this operating unit until then can be associated with another operating unit.

In an especially preferred embodiment of the invention, the electronic parcel box system is configured in such a way that the central control unit has an interface for receiving information from a database server that contains filling data.

In this way, the electronic parcel box system can be flexibly adapted to operating situations and user habits.

For example, if an operating unit malfunctions, the parcel boxes that had been associated with this operating unit until then can be associated with one or more other operating units. This, too, enhances the flexibility to adapt to the needs or habits of users of the parcel box system.

Especially preferred embodiments of the invention are characterized in that they allow internal communication between the central control unit and the operating units as well as external communication between the central control unit and one or more database servers.

In an advantageous embodiment of the invention, such an integration of the data communication takes place in such a way that the central control unit has an interface for transmitting information to a database server that has been provided with filling data.

Another enhancement in the flexibility and versatility of the electronic parcel box system can be achieved in that the central control unit has a computer that can store and process operating data and/or the filling status of the parcel boxes.

The term "computer" is not to be construed in any limiting manner. Its refers to any unit that is capable of executing computations, for example, a work station, a personal computer, a microcomputer or a circuit that is suitable for executing computations and/or comparisons.

Moreover, it is advantageous to equip the central control unit with additional interfaces for checking and/or changing operating states. This makes the electronic parcel box system particularly easy to maintain.

Another subject matter of the invention is a logistic system that is characterized in that it contains several electronic parcel box systems networked with each other.

Such a logistic system constitutes an especially preferred way to achieve the objective according to the invention, namely, allowing a flexible delivery and pick-up of parcels from parcel boxes.

By networking several electronic parcel box systems, it is especially possible to deliver mailpieces in a particularly flexible manner.

Thus, for example, if all of the parcel boxes of a given electronic parcel box system are occupied, it is possible to transport additional mailpieces to another electronic parcel box system and to keep them ready for pick-up there.

By integrating a preferably centrally operated database server, parcel shipments can be delivered to the electronic parcel box systems as a function of the filling status of the electronic parcel box system and/or of the parcel boxes contained therein.

Additional advantages, special features and practical refinements of the invention ensue from the subclaims and from the following presentation of preferred embodiments of the invention making reference to the drawings.

The drawings show the following:

Figure 1 the structure of a delivery machine interface that is suitable as a communication platform with the electronic parcel box system;

Figure 2 a schematic diagram of components used for user registration, user assistance and shipment tracking;

Figure 3 a schematic diagram of function procedure calls for the implementation of in-house service functions;

Figure 4 a schematic diagram of interfaces employed for user notification;

Figure 5 a detailed view of interfaces used for the filling and administration of parcel boxes;

Figure 6 a detailed view of interfaces used for the configuration of delivery machines and the capacity utilization, and

Figure 7 process steps for correcting errors.

The embodiment presented below shows the integration of the electronic parcel box system into a comprehensive logistic system.

Such an integration of the electronic parcel box system into a comprehensive system is especially advantageous since in such a case, the flexibility attained with the set-up of the electronic parcel box systems according to the invention allows them to be used especially efficiently.

Although such a use of the parcel box systems in the comprehensive logistic system is especially preferred, as a matter of principle, the electronic parcel box systems can also be operated without being networked. In these embodiments as well, the integration of the central control unit for controlling the operating units makes the electronic parcel box systems more flexible to use.

However, the application possibilities for the invention are even further augmented by networking multiple parcel box systems with each other and/or with the preferably centrally operated database server.

When several of these advantageous components are integrated, the result is a comprehensive system having the following constituents, which are also referred to below as Post24 since it is possible to achieve automated, round-the-clock operation:

1. Multiple parcel dispensing units that preferably each have at least one operating unit.

2. A central control unit with which the individual delivery machines are networked in clusters and which monitors the operating states and functions of the delivery machines. The central control unit serves as a "concentrator" for the parcel transactions that it forwards to the Post24 server. Moreover, it distributes the operating instructions of the Post24 server to the parcel delivery machines.

3. The Post24 server that administers the customer and parcel data and that registers and controls the procedures of the delivery machine and that operates the central program applications (registration, CallCenter, Track&Trace, etc.).

The system has a modular design. A standardized interface is defined that regulates the communication between the delivery machine, or the CallCenter, and the Post24 server. In this manner, delivery machines made by different manufacturers can be incorporated.

The modular design of the individual parcel box systems is especially well-suited in terms of increasing the flexibility and versatility of the individual electronic parcel box systems as well as of the comprehensive logistic system.

Preferably, the electronic parcel box systems used in the logistic system are structured in such a way that they execute the functions and operating contents of the comprehensive logistic system in an especially effective and efficient manner.

As a matter of principle, the electronic parcel box systems can be implemented in the form of numerous different technical solutions.

The embodiments of electronic parcel box systems presented below, however, are particularly well-suited in view of their reliability and user-friendliness.

One of the two systems is characterized in that the parcel boxes can each be individually closed with a door. Another, likewise preferred, configuration of the embodiment of the parcel box system is characterized by a cyclic elevator system with special transport mechanisms for individual boxes for purposes depositing and removing the individual parcels.

In the embodiments shown, the delivery machine allows loading and picking-up of parcels or of contents via differently sized parcel boxes. An especially preferred feature of the delivery machine system is that parcel boxes are only temporarily associated with

certain recipients, namely, only when the box contains something for that specific recipient. For this purpose, first of all, the potential recipient should be known to the delivery machine system and secondly, the shipments should be administered centrally and so as to be recipient-related.

During the loading procedure, a parcel identification code and a recipient identification code are detected. On the basis of the recipient identification code, the matching access code is requested from the Post24 server (central system). When the parcel is picked up, the person doing so identifies himself with his personal recipient identification code, for example, using a magnetic card and the secret personal access code. If this information matches the information stored in the delivery machine, the contents of a box are released to the recipient.

If it is a COD parcel or if a sum of money has to be collected when the parcel is handed over, then the amount is first displayed and the person picking it up can pay this amount via a payment terminal (EC card, cash card, etc.).

Parcels (or goods) that are being returned by recipients are detected by the delivery machine and reported to the Post24 server, which initiates the pick-up procedure by the appropriate logistics personnel.

Advantageously, the central control unit – which is also called a control panel – has the properties described below.

The individual delivery machines are networked via the central control unit that clusters and combines the communication ("concentrator"). Service and maintenance tasks are monitored via the central control unit. The central control unit recognizes whether delivery machines or components of the delivery machine (operating unit, printer, individual boxes) are ready for operation and which operating events are taking place in the various delivery machines.

The Post24 server is a data server that is networked via Internet-based web technology with the delivery machines, or with the control panel. The Post24 server is an interface to additional database applications. These database applications allow, for example, an integration of invoicing and shipment tracking functions.

The delivery machine is preferably only used by recipients who have previously signed up with the Deutsche Post for this service and whose data is stored on the Post24 server. The recipients who are thus registered receive a recipient identification code that they are supposed to use as part of their address information and they also receive a personal secret access code.

Once a delivery has been made to the delivery machine, the recipient is automatically notified by the Post24 server. The notification is preferably given electronically, for example, by sending an SMS or an e-mail. However, it is likewise possible for a letter to be automatically printed and sent to the recipient. The recipient can then pick up the parcel at the delivery machine at his own convenience. If this is not done within a certain period of time, a reminder is sent and, once the delivery period has lapsed, the parcel is made ready to be returned to the sender.

The parcel pick-up system is suitable for various areas of application. On the one hand, this delivery machine system can deliver normal parcels for final customers, but on the other hand, it can also meet special requirements of business customers.

Thus, a fixed or variable portion of the delivery machine parcel boxes can be rented for business partners and customers of the logistic company employing the system so that these customers can use the system accordingly. The delivery machine system is thus a central depot for these customers and the stock of goods and movement of goods in this depot can be controlled centrally and invoiced accordingly.

In particular, this is a system consisting of centralized and decentralized components in order to achieve an intelligent, demand-based ordering and delivery system.

The system comprises various components and interfaces that effectively and flexibly achieve a logistic system with delivery and service capabilities that have not been possible up until now.

Examples of functions that are integrated into the system are:

- parcel data management
- user data management
- registration of customers
- B2B partner management
- tracking of packages
- management of the delivery machine configuration
- capacity planning
- notification
- statistical data evaluation
- delivery machine interface
- in-house service

It is advantageous to make a distinction between basic management components and the access to the parcel data and user data as well as to secondary components that map the business processes 4 into the system.

The modular structure of the system allows an independent incremental development of the various services.

Since an especially preferred embodiment of the system according to the invention and of its components, likewise according to the invention, is based on an implementation of EJBs (Enterprise JavaBeans) technology, the basic components are mapped onto parcel EJBs and User EJBs. All other components communicate with these EJBs when they have to request or change parcel and customer data.

The secondary components are, in part, complex partial systems that are made up of classes, EJBs and graphical user interfaces (GUIs). In order to allow communication of the components that is independent of the component design, each component provides a binding interface to its services in the form of a facade object – or EJB. The interfaces of the various components are described below.

Delivery machine interface:

The delivery machine interface is the communication platform with the delivery machines. The interface receives requests from the delivery machine and forwards them to the pertinent components. If the delivery machine expects data in response to its request, then this data is returned to the interface by the components and subsequently forwarded to the delivery machine.

Figure 1 shows the structure of the components. An XML parser carries out the mapping of the XML files that are sent per http post into the function procedure calls that can be interpreted by the logistic system of the Machine Interface class, evaluates the function procedure calls and forwards them to the pertinent components, namely, User, Parcel, DeliveryMachine and Notification.

The components User Registration, B2B Partner Management and Parcel Tracking are preferably encapsulated in a CallCenter facade. The facade takes over the communication with the EJBs for user and parcel data management, that is to say, User and Parcel. The component makes available web-based GUIs for registering and parcel tracking. These are encoded in JSP. Actions by the users are translated by means of a Worker Bean into function procedure calls and forwarded to the CallCenter facade. Preferably, it is the CallCenter facade that only now takes over the execution of the business logic circuit. The graph presented below provides a rough overview of this scenario.

The In-House Services component encapsulates the services provided by the logistic system for the in-house service into the delivery bases. At the interface to the employees of the in-house service, there is a JSP that implements the GUI. Actions by the in-house employees are translated by means of a Worker Bean into function procedure calls and forwarded to the In-House Service facade. The In-House Service facade thus forms the interface to the actual function logic circuit of the component.

The In-House Service facade also constitutes the communication interface to the other components of the system.

In order to map the required function logic circuit, the component needs access to the basic components, namely, User and DeliveryMachine.

The structure of the component is shown in the figure below.

Another component that, thanks to the modular structure of the logistic system, can be installed without any problems and that advantageously improves the logistic system and the components used therein, especially the parcel box system, is a notification component.

The notification component effectuates the sending of messages to the user of the logistic system. The component preferably has two tasks. It maps the notification regimen of the logistic partners and takes over the technical transmission of messages to the devices of the customers.

The interface to the other components of the system is implemented by a facade class called NotificationFacade. This class provides numerous methods that serve to inform the component about events that can trigger a notification being sent to a user. At the present time, such events are triggered by the interface component as the result of an action at the delivery machine and of the registration component. In order to be able to configure the target and the content pertaining to the message, the component needs to

have access to address data of the message recipient. All of the message recipients managed in this especially preferred embodiment of the logistic system should thus make available the Notifiable interface. These are, for example, the EJBs LogisticPartnerClient, User and LogisticPartners.

The organization of the user and parcel data can be carried out in many different ways. A user and parcel management system in the form of an object-oriented model (object model) that can be implemented especially advantageously will be presented below.

The central object of the object model for the user management is the EntityBean UserEJB. This object stores all data on persons who are permitted to operate the delivery machine. Final customers as well as deliverers and service personnel are classified as users. A complex object model (person as basic class, derived classes for the various roles of the person) was dispensed with.

This yields the preferred embodiment, namely, to place the role of the user into the object as an attribute. The role is defined as int. In the Const class, expressive variable names are defined, for example,:

```
public class Const {
        ...
        static final int ROLE_DPAG_RECIPIENT = 1;
        ...
}
```

Moreover, the user object should have a reference to the associated LogisticPartners or LogisticPartnerClient. LogisticPartners are, for example, the postal service provider that operates the logistic system, or other companies that are given access to components of the logistic system, including the loading of the parcel boxes. LogisticPartnerClients are the customers of a LogisticPartner, that is to say, for example, Telekom as a customer of Danzas.

Depending on the role of the user, the object receives a reference to a LogisticPartner (ROLE_B2B_DELIVERY_AGENT) or to a LogisticPartnerClient (ROLE_B2B_RECIPIENT). In each case, the other reference remains empty (It is never permissible for both references to be filled in at the same time since a person is either delivering or picking up a shipment, never both).

The UserGroupEJB object (likewise an EntityBean) allows the combination of several users into one group. In addition to a list of users, the UserGroup has a reference to the LogisticPartner or to the LogisticPartnerClient to which all members of the groups belong. The redundant storage of the reference in the User as well as in the UserGroup simplifies the access to information in the LogisticPartner(Client).

Since parcels can be sent to both types (User and UserGroup), the two objects implement the Interface Addressable. The interface has the following methods: getUsers( ), getID( ) and getExpiryMinutes( ). This ensures that parcels that are sent to an addressable can be processed by the server.

The getUsers( ) method yields a list of user EJBs. If it is a UserGroup, this list contains several EJBs, and if it is a User, only one EJB is present.

The getID( ) method returns the ID of the User or of the UserGroup. Since the addressable has to be able to be permanently stored, the subordinate object can be found through this ID. This results in the request that the IDs for the User and UserGroup be taken from a set of numbers. Moreover, in the object Post24ServerProperties, there is the newUserID( ) method. It returns an unambiguous continuous ID. Internally, this method accesses a database application, especially an Oracle sequence.

The getExpiryMinutes( ) method in the addressable is needed in order to determine the ExpiryDate of the parcel. The permitted retention time of a parcel depends on the client. In order to calculate the ExpiryDate, the getExpiryMinutes( ) method in both objects

User and UserGroup accesses the LogisticPartnerClient.getExpiryMinutes( ) method and adds this permissible retention time to the current point in time.

The User and the UserGroup are created by the registration or by the administration tool.

The EntityBean Parcel is provided in order to manage the parcels. The ParcelID is issued by the delivery machine and the server is notified via the interface with notifyDelivery( ). Then the delivery machine facade creates a new Parcel Object with the ParcelID as the Primary Key.

The recipient (transmitted in the form of a customer number) is stored as addressable. In addition to the recipient, there is the substitute attribute, likewise of the addressable type. It cannot be set at notifyDelivery( ) but rather at a later point in time by calling the substituteBy method (addressable Addressable). The substitute arrangement is mapped by means of this attribute. The Parcel Object offers the following methods: getRecipients( ) and getSubstitutes( ), each of which returns a list of User Objects.

In generating a new Parcel Object, for example, by means of the notifyDelivery( ) delivery machine interface, the ExpiryDate is calculated using the above-mentioned method and the notification is made to the Parcel Object. Thus, the create method contains the ExpiryDate. Each LogisticPartner can have different calculation rules for the ExpiryDate.

In the case of COD parcels, a COD object is dispensed, while packages and parcels without COD are generated by a create method without COD. In the case of parcels, the IdentCode of the parcel is transferred in the create method.

In addition, the Parcel Object is informed about which LogisticPartner it belongs to. This information is necessary for the capacity planning.

The history is maintained by the database using a trigger that transmits the old data record to an archive table before changing a data record. The getHistory( ) method provides a listing of the status of all archive entries.

The delivery machine configuration and the capacity management are preferably likewise integrated into the logistic system as modular components.

The delivery machines of the Post24 system make available a fixed number of boxes of different sizes for storage. The boxes are used by different logistic companies. Since the allocation of the delivery machine boxes among the logistic companies should be known for purposes of the parcel handling as well as for the cost calculation, delivery machine configurations are administered in an EJB DeliveryMachine. The EJB allows the setting and querying of the box configurations of each delivery machine as well as the calculation of the current and future filling with parcels.

The actual box configuration, that is to say, the number of boxes rented by a logistic partner broken down by box size, is represented by two EJBs BoxTypeAssignment and BoxType. The DeliveryMachine allocates each logistic partner a certain amount of BoxTypeAssignment EJBs, namely, one for each rented box size. Each BoxTypeAssignment stores the type and number of rented boxes. The box size is stored in its own EJB with the name BoxType.

Due to the modular structure of the parcel box system used as well as of the logistic system, additional components can be integrated at any point in time. For example, it is possible to provide additional components for evaluation functions. Examples of such a data evaluation functionality are evaluations of the parcel data and customer data for statistical purposes.

Protocolling (logging) is advantageous, especially for checking the proper functioning.

The logging classes provided by the application server are used to log errors that have occurred. These are encapsulated for greater usefulness.

The class de.Post24.util.P24Log provides a statistical logging method:

```
public static void log  (int severity,
                         java.lang.String system,
                         Java.lang.String msg,
                         java.lang.Throwable t)
```

The meanings here are:

severity	severity of the error. Four error classes are distinguished, which can be called up via constants within the P24Log class: P24Log.debug, P24Log.info, P24Log.warning, P24Log.error
system	name of the partial system
msg	error text to be stored
t	exception

Simpler "lean" versions of this method are likewise possible, preferably by replacing certain parameters by predefined values, for example:

severity	P24Log.debug
system	name of the partial system
t	not applicable

Using the techniques shown, an adaptation of the configuration parameters can be done quickly and reliably.

The person skilled in the art can replace embodiments used in this presentation by substituting functionalities of other programming languages and program functions.

In the preferred programming language java, it is also possible to use other functionalities than the ones mentioned.

In particular, it is advantageous for the person skilled in the art to use the process steps known from the other documents

http://java.sun.com/docs/codeconv/
Java Coding Conventions

for the implementation of additional functions and for modifying and improving the functions employed.

Patent Claims:

1. An electronic parcel box system, **characterized in that** it comprises a plurality of electronic parcel boxes, whereby in each case, several electronic parcel boxes are associated with an operating unit, and whereby a central control unit is provided for controlling the operating units.

2. The electronic parcel box system according to Claim 1, **characterized in that** the operating units comprise a control means for controlling the opening and/or the closing of the parcel boxes associated with the operating units.

3. The electronic parcel box system according to Claim 2, **characterized in that** the central control unit has a means for associating the parcel boxes with the operating units.

4. The electronic parcel box system according to Claim 3, **characterized in that** the association of the parcel boxes with the operating units is variable.

5. The electronic parcel box system according to one or more of the preceding claims, **characterized in that** the central control unit has an interface for receiving information from a database server that contains filling data.

6. The electronic parcel box system according to one or more of the preceding claims, **characterized in that** the central control unit has an interface for transmitting information to a database server that has been provided with filling data.

7. The electronic parcel box system according to one or more of the preceding claims, **characterized in that** the central control unit has a computer that can store and process operating data and/or the filling status of the parcel boxes.

8. A logistic system, **characterized in that** it contains several electronic parcel box systems networked with each other.

9. The logistic system according to Claim 8, **characterized in that** it has a database server.

10. The logistic system according to Claim 9, **characterized in that** the database server has means for storing and/or transmitting filling data relating to the parcel box systems.

FIG 1

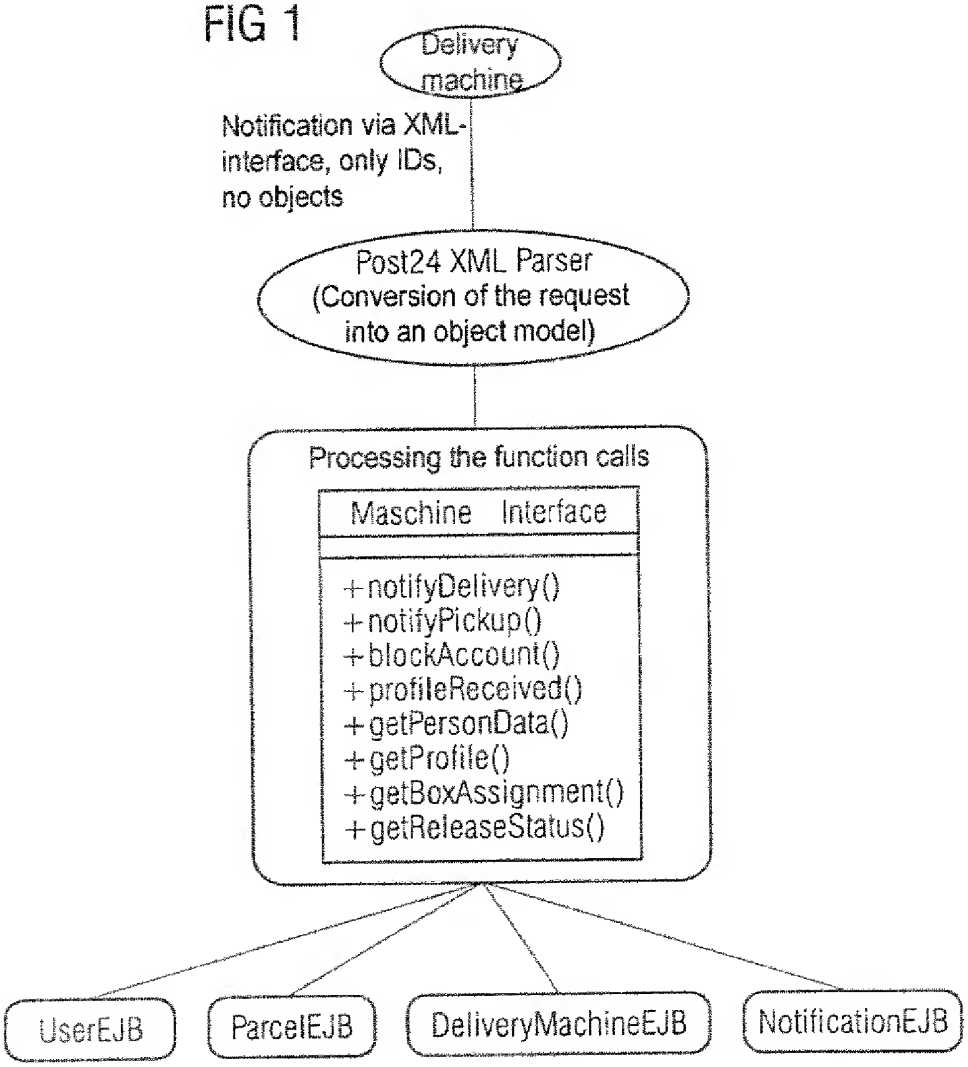

Delivery machine
Notification via XML-interface, only IDs, no objects
Post24 XML Parser (Conversion of the request into an object model)
Processing the function calls
Maschine Interface
+notifyDelivery()
+notifyPickup()
+blockAccount()
+profileReceived()
+getPersonData()
+getProfile()
+getBoxAssignment()
+getReleaseStatus()
UserEJB
ParcelEJB
DeliveryMachineEJB
NotificationEJB

FIG 2

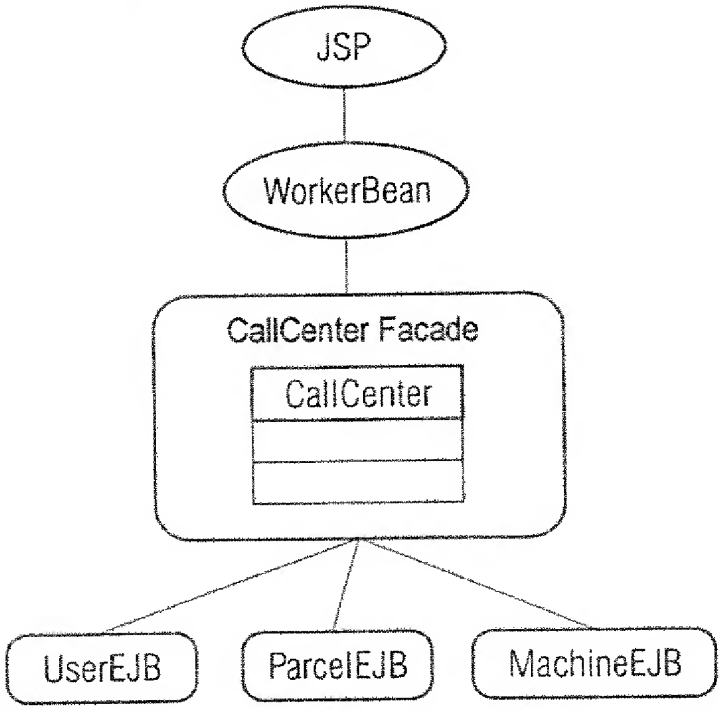

JSP
WorkerBean
CallCenter Facade
CallCenter
UserEJB
ParcelEJB
MachineEJB

FIG 3

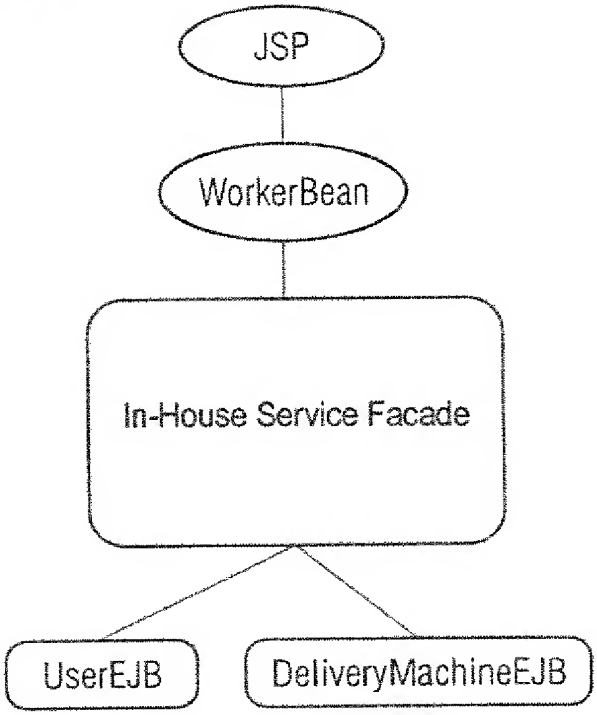

JSP
WorkerBean
In-House Service Facade
UserEJB
DeliveryMachineEJB

FIG 4

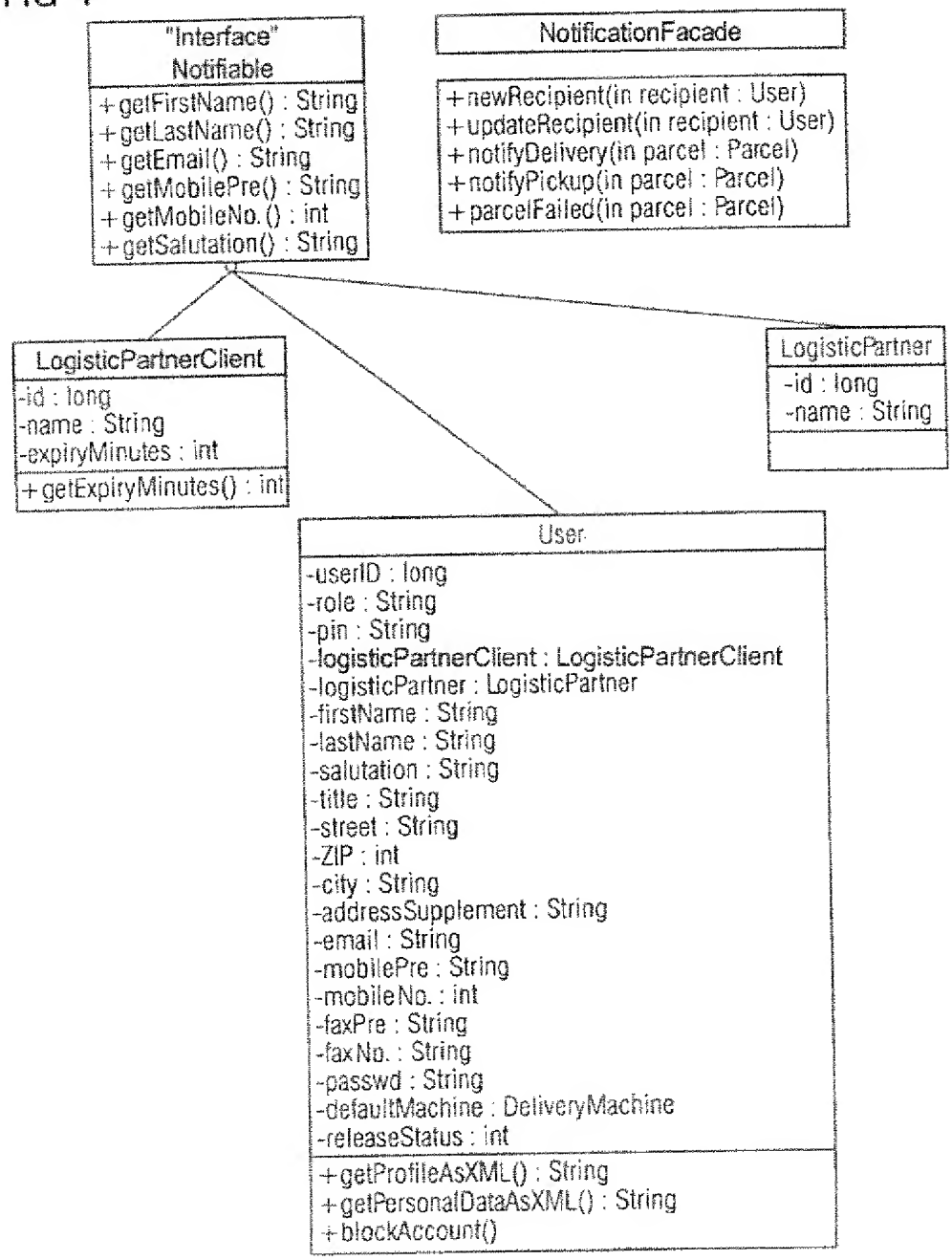

"Interface"
Notifiable
+getFirstName() : String
+getLastName() : String
+getEmail() : String
+getMobilePre() : String
+getMobileNo. () : int
+getSalutation() : String
NotificationFacade
+newRecipient(in recipient : User)
+updateRecipient(in recipient : User)
+notifyDelivery(in parcel : Parcel)
+notifyPickup(in parcel : Parcel)
+parcelFailed(in parcel : Parcel)
LogisticPartnerClient
-id : long
-name : String
-expiryMinutes : int
+getExpiryMinutes() : int
LogisticPartner
-id : long
-name : String
User
-userID : long
-role : String
-pin : String
-logisticPartnerClient : LogisticPartnerClient
-logisticPartner : LogisticPartner
-firstName : String
-lastName : String
-salutation : String
-title : String
-street : String
-ZIP : int
-city : String
-addressSupplement : String
-email : String
-mobilePre : String
-mobileNo. : int
-faxPre : String
-faxNo. : String
-passwd : String
-defaultMachine : DeliveryMachine
-releaseStatus : int
+getProfileAsXML() : String
+getPersonalDataAsXML() : String
+blockAccount()

FIG 5

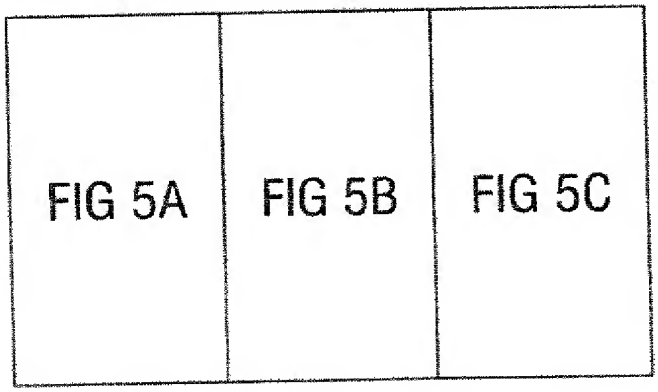


FIG 5A

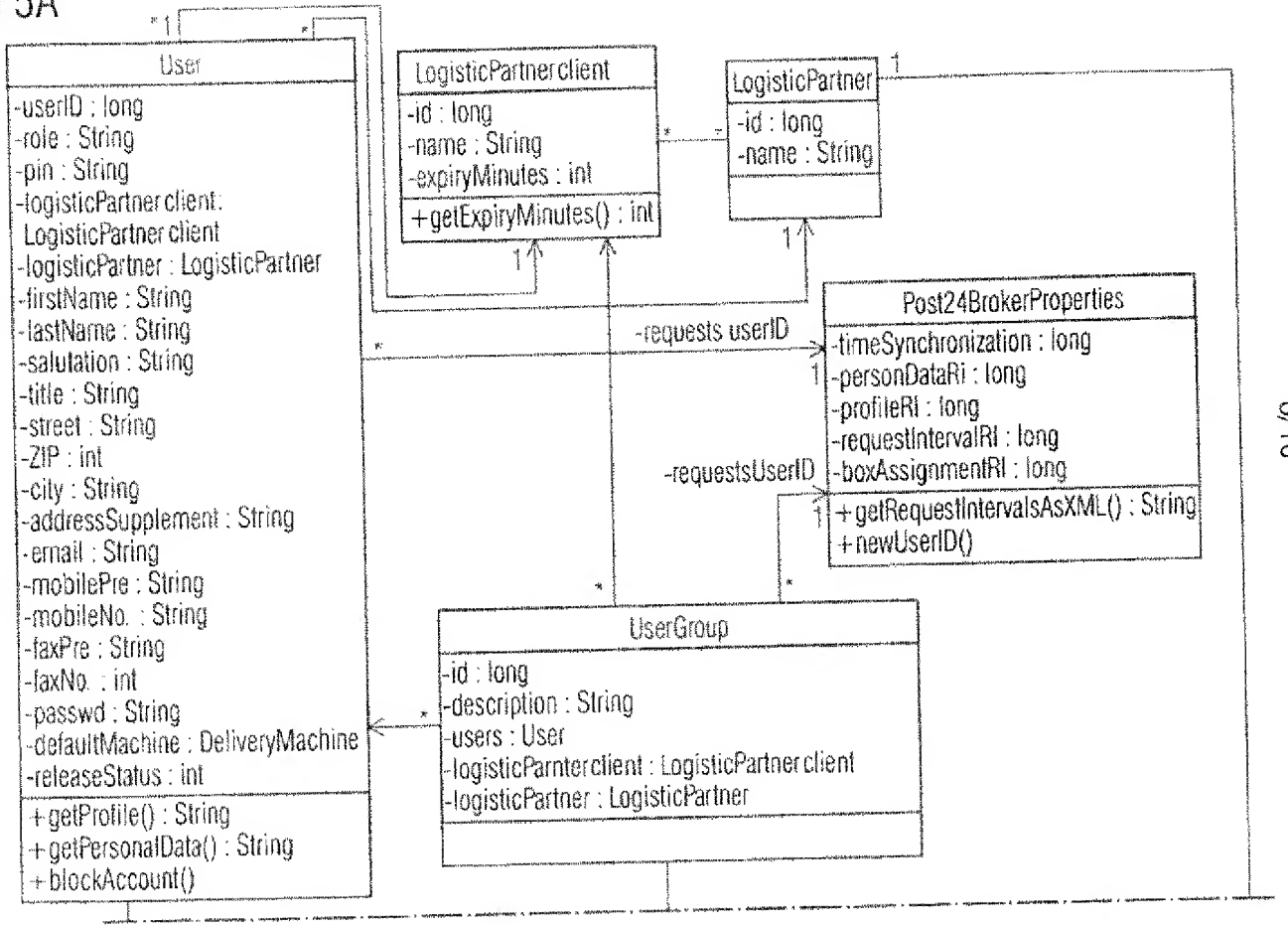

User
-userID : long
-role : String
-pin : String
-logisticPartnerclient:
LogisticPartnerclient
-logisticPartner : LogisticPartner
-firstName : String
-lastName : String
-salutation : String
-title : String
-street : String
-ZIP : int
-city : String
-addressSupplement : String
-email : String
-mobilePre : String
-mobileNo. : String
-faxPre : String
-faxNo. : int
-passwd : String
-defaultMachine : DeliveryMachine
-releaseStatus : int
+getProfile() : String
+getPersonalData() : String
+blockAccount()
LogisticPartnerclient
-id : long
-name : String
-expiryMinutes : int
+getExpiryMinutes() : int
LogisticPartner
-id : long
-name : String
-requests userID
-requestsUserID
Post24BrokerProperties
-timeSynchronization : long
-personDataRI : long
-profileRI : long
-requestIntervalRI : long
-boxAssignmentRI : long
+getRequestIntervalsAsXML() : String
+newUserID()
UserGroup
-id : long
-description : String
-users : User
-logisticParnterclient : LogisticPartnerclient
-logisticPartner : LogisticPartner

FIG 5B

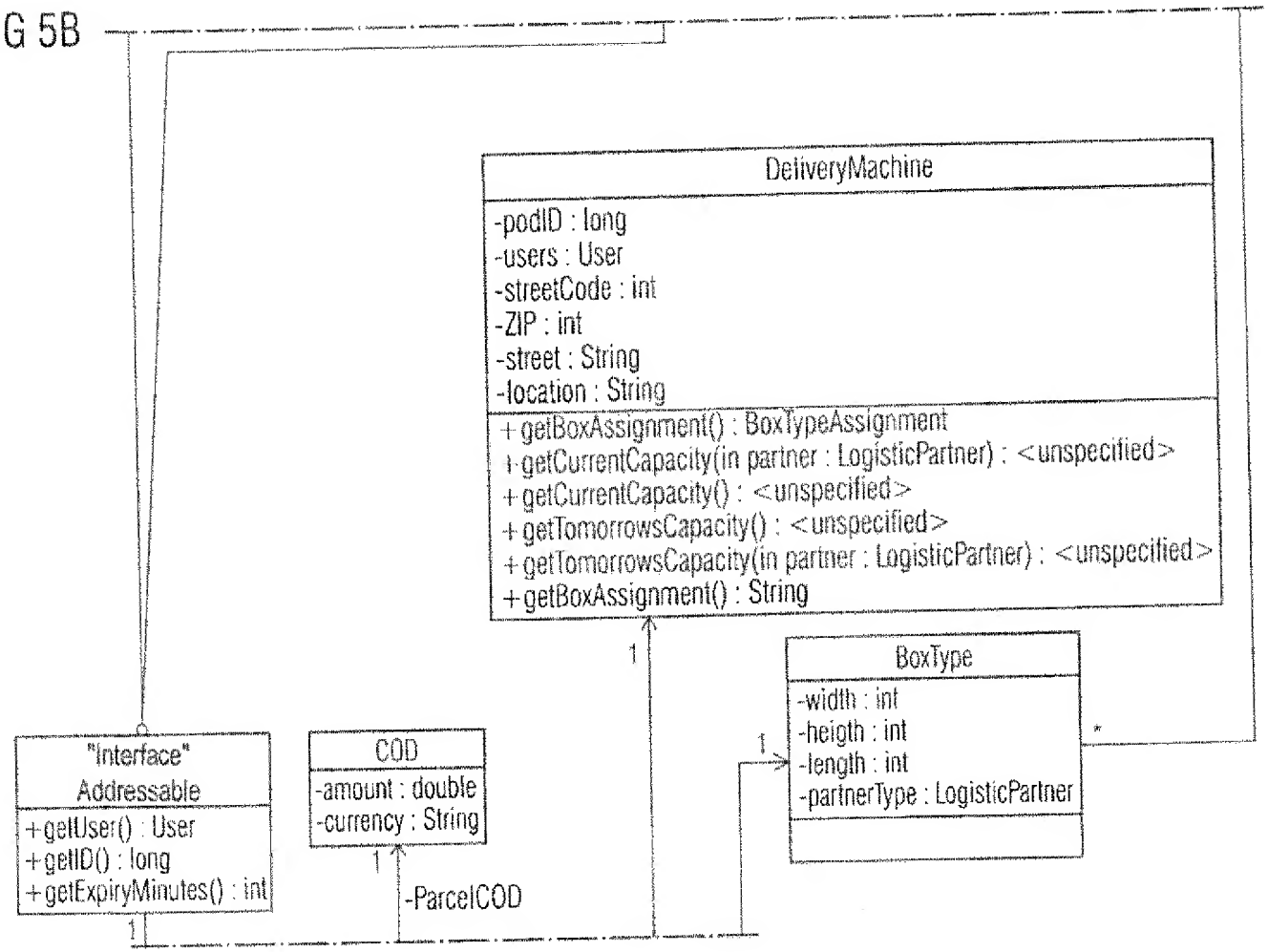

DeliveryMachine
-podID : long
-users : User
-streetCode : int
-ZIP : int
-street : String
-location : String
+getBoxAssignment() : BoxTypeAssignment
+getCurrentCapacity(in partner : LogisticPartner) : <unspecified>
+getCurrentCapacity() : <unspecified>
+getTomorrowsCapacity() : <unspecified>
+getTomorrowsCapacity(in partner : LogisticPartner) : <unspecified>
+getBoxAssignment() : String
BoxType
-width : int
-heigth : int
-length : int
-partnerType : LogisticPartner
"Interface"
Addressable
+getUser() : User
+getID() : long
+getExpiryMinutes() : int
COD
-amount : double
-currency : String
-ParcelCOD
1
*

FIG 5C

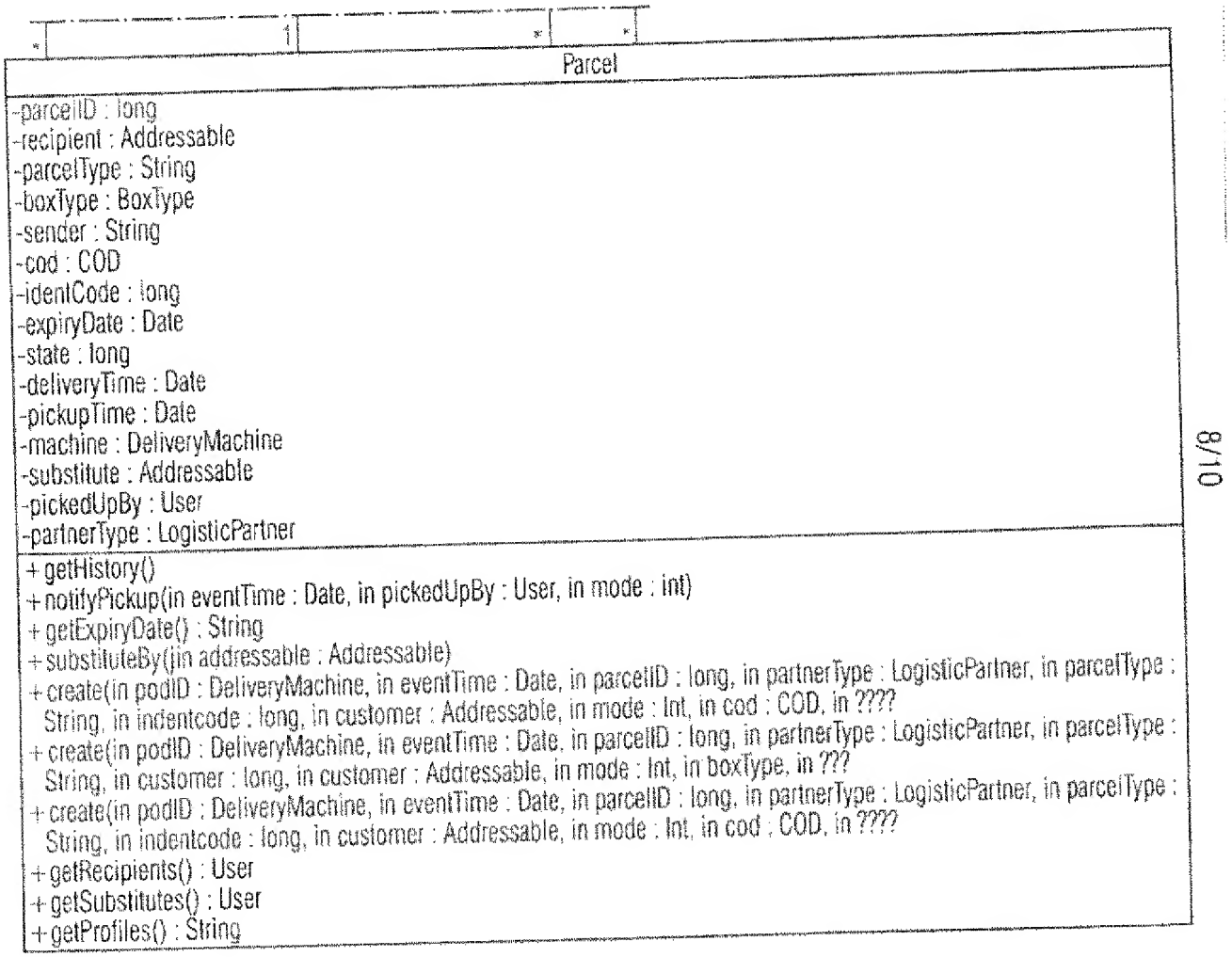

*
1
*
*
Parcel
-parcelID : long
-recipient : Addressable
-parcelType : String
-boxType : BoxType
-sender : String
-cod : COD
-identCode : long
-expiryDate : Date
-state : long
-deliveryTime : Date
-pickupTime : Date
-machine : DeliveryMachine
-substitute : Addressable
-pickedUpBy : User
-partnerType : LogisticPartner
+ getHistory()
+ notifyPickup(in eventTime : Date, in pickedUpBy : User, in mode : int)
+ getExpiryDate() : String
+ substituteBy(in addressable : Addressable)
+ create(in podID : DeliveryMachine, in eventTime : Date, in parcelID : long, in partnerType : LogisticPartner, in parcelType : String, in indentcode : long, in customer : Addressable, in mode : Int, in cod : COD, in ????
+ create(in podID : DeliveryMachine, in eventTime : Date, in parcelID : long, in partnerType : LogisticPartner, in parcelType : String, in customer : long, in customer : Addressable, in mode : Int, in boxType, in ???
+ create(in podID : DeliveryMachine, in eventTime : Date, in parcelID : long, in partnerType : LogisticPartner, in parcelType : String, in indentcode : long, in customer : Addressable, in mode : Int, in cod : COD, in ????
+ getRecipients() : User
+ getSubstitutes() : User
+ getProfiles() : String

FIG 6

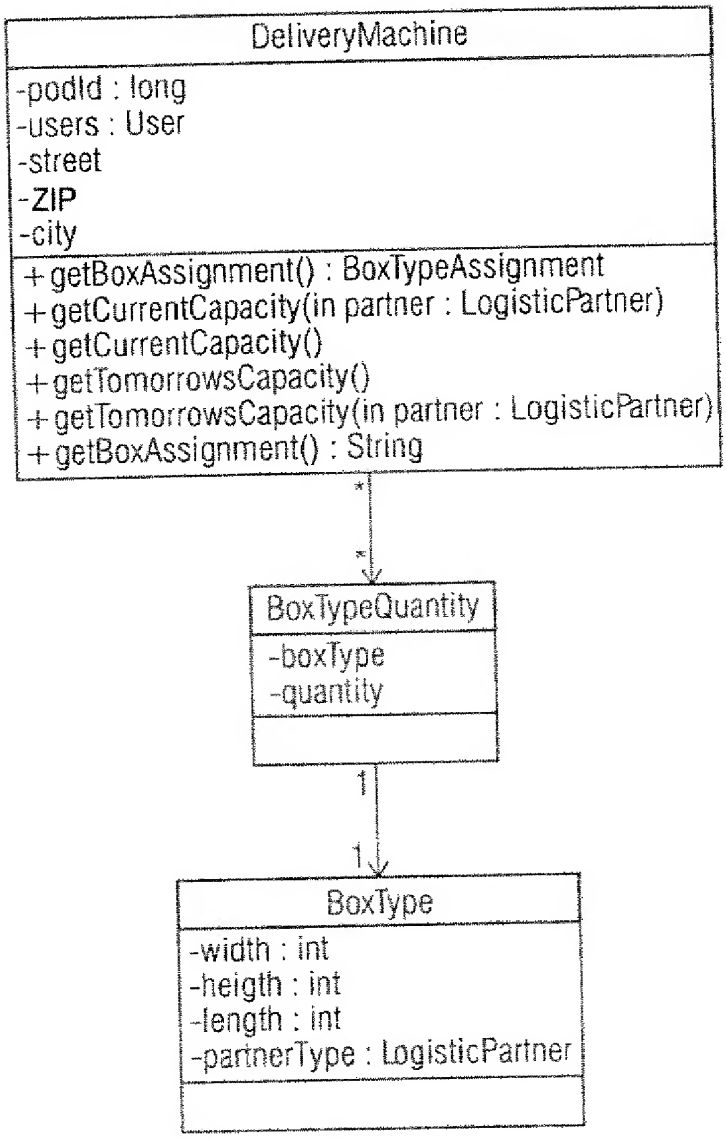
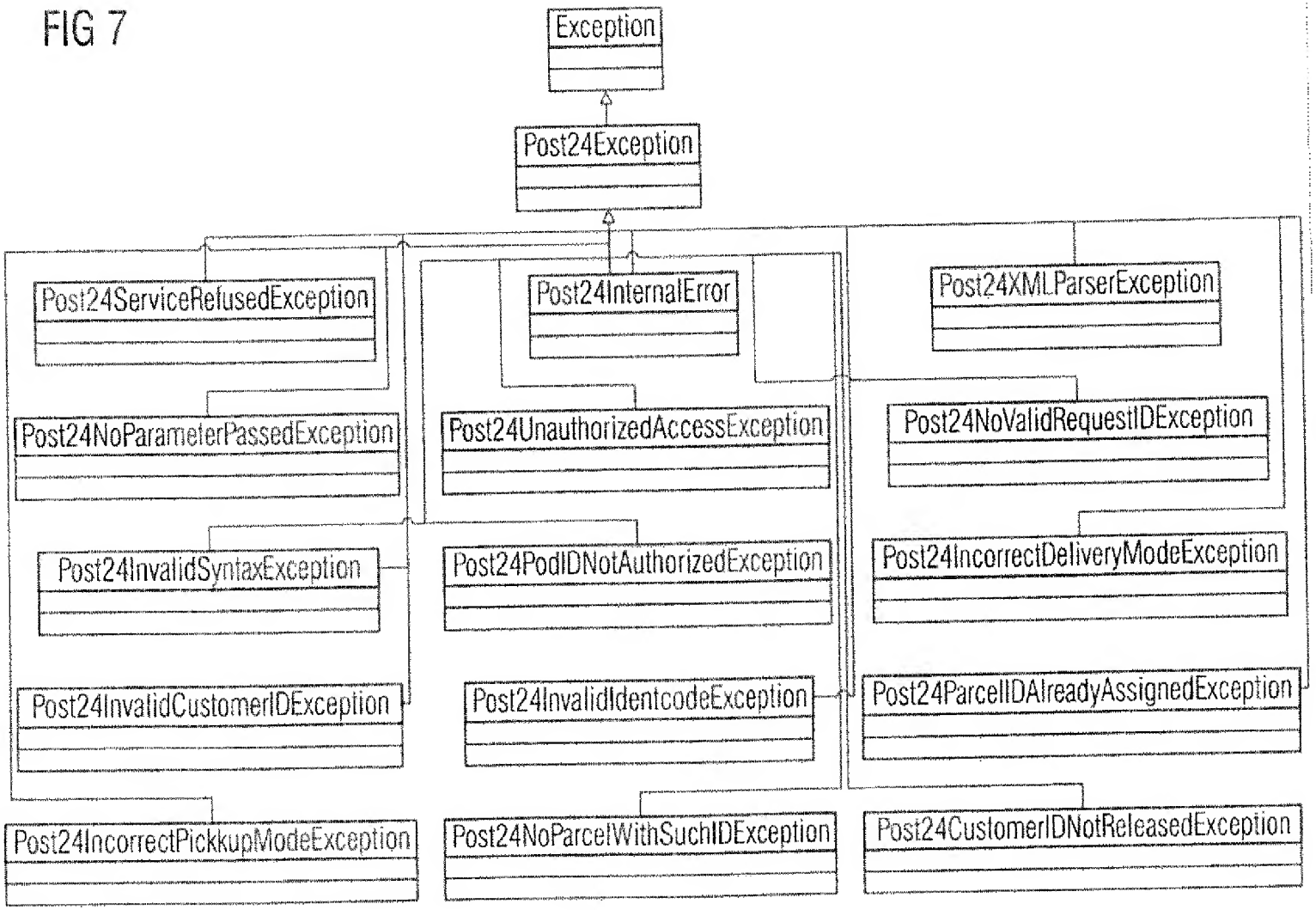

DeliveryMachine
-podId : long
-users : User
-street
-ZIP
-city
+getBoxAssignment() : BoxTypeAssignment
+getCurrentCapacity(in partner : LogisticPartner)
+getCurrentCapacity()
+getTomorrowsCapacity()
+getTomorrowsCapacity(in partner : LogisticPartner)
+getBoxAssignment() : String
*
*
BoxTypeQuantity
-boxType
-quantity
1
1
BoxType
-width : int
-heigth : int
-length : int
-partnerType : LogisticPartner

FIG 7


Exception
Post24Exception
Post24ServiceRefusedException
Post24InternalError
Post24XMLParserException
Post24NoParameterPassedException
Post24UnauthorizedAccessException
Post24NoValidRequestIDException
Post24InvalidSyntaxException
Post24PodIDNotAuthorizedException
Post24IncorrectDeliveryModeException
Post24InvalidCustomerIDException
Post24InvalidIdentcodeException
Post24ParcelIDAlreadyAssignedException
Post24IncorrectPickkupModeException
Post24NoParcelWithSuchIDException
Post24CustomerIDNotReleasedException